Exercise Activity Prediction Model

Ling Kok Heng, date: 22 March 2015

EXECUTIVE SUMMARY

Using devices such as Jawbone Up, Nike FuelBand, and Fitbit it is now possible to collect a large amount of data about personal activity relatively inexpensively. These type of devices are part of the quantified self movement - a group of enthusiasts who take measurements about themselves regularly to improve their health, to find patterns in their behavior, or because they are tech geeks. One thing that people regularly do is quantify how much of a particular activity they do, but they rarely quantify how well they do it.

The aim of this report was to predict how well the participants performed in lifting the barbells in 5 different ways. The datasets were collected from accelerometers placed on the belt, forearm, arm, and dumbell of all six participants.

PREPROCESSING OF DATA

```
## Loading required package: lattice
## Loading required package: ggplot2
## randomForest 4.6-10
## Type rfNews() to see new features/changes/bug fixes.
```

Both the training (pml-training.csv) and testing (pml-testing.csv) files were directly downloaded into a created file folder (./pmldata) in the working directory (see Appendix, Data Sources for more detail R-codes).

```r
# read the training csv file
pml_training <- read.csv("./pmldata/pml-training.csv", na.strings= c("NA",""," "))
```

The training dataset, pml-training.csv, was then loaded into R and was inspected to contain lots of “NA” values.The “NA” values in the training datasets would cause noises for the model. Hence, these columns containing “NA” were removed from the data set. The first eight columns that acted as identifiers for the experiment were also removed.

```r
# removing columns with NAs
pml_training_NAs <- apply(pml_training, 2, function(x) {sum(is.na(x))})
pml_training_clean <- pml_training[,which(pml_training_NAs == 0)]

# removing identifier columns such as name, timestamps, etc
pml_training_clean <- pml_training_clean[8:length(pml_training_clean)]
```

CREATING THE MODEL

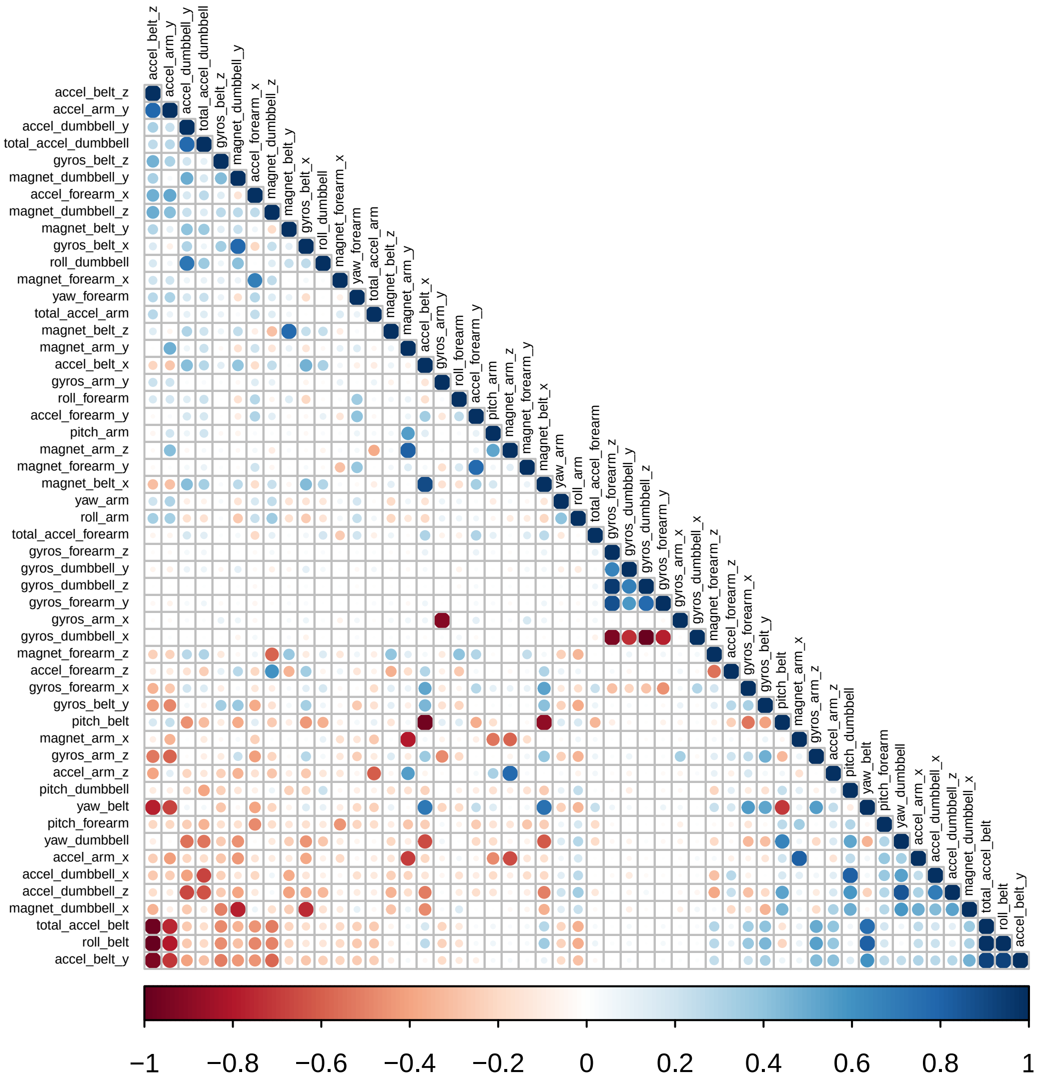
Training the Model

The training dataset was splitted up into training and cross validation sets in a 70:30 ratio in order to train the model. It will also be used to test against data that was not specifically fitted to.

```r
# split the cleaned testing data into training and cross validation, 70:30 ratio
inTrain <- createDataPartition(y = pml_training_clean$classe, p = 0.7, list = FALSE)
training_set <- pml_training_clean[inTrain, ]
crossval_set <- pml_training_clean[-inTrain, ]
```

The **Random Forest model** was selected to predict the classification because it has methods for balancing error in class population unbalanced data sets. The correlation between any two trees in the forest increases the forest error rate. Therefore, a correlation plot was produced so as to determine how strong the variables' relationships are among each other.

```r
# plot a correlation matrix
correlMatrix <- cor(training_set[, -length(training_set)])
corrplot(correlMatrix, order = "FPC", method = "circle", type = "lower", tl.cex = 0.45,  tl.col = rgb(0
```



The dark red and blue colours within the plot indicated a highly negative and positive relationships respectively between the variables. The highly correlated predictors means that all of variables can be included in the model.

The model was fitted with the outcome set to the training classe with all the other variables as the predictor.

```r
# fitting the model to predict the classe with everything else as a predictor
model <- randomForest(classe ~ ., data = training_set)
model
```

```
##
```

```
## Call:
##  randomForest(formula = classe ~ ., data = training_set)
##                Type of random forest: classification
##                      Number of trees: 500
## No. of variables tried at each split: 7
##
##         OOB estimate of  error rate: 0.5%
## Confusion matrix:
##      A    B    C    D    E  class.error
## A 3903    1    1    0    1 0.0007680492
## B   12 2638    8    0    0 0.0075244545
## C    0   17 2376    3    0 0.0083472454
## D    0    0   16 2235    1 0.0075488455
## E    0    0    1    8 2516 0.0035643564
```

The model produced a very small OOB error rate of 0.5%. This was deemed satisfactory enough to progress the testing.

Cross-validating the Model

The model was used to classify the remaining 30% of the datasets. The results were placed in a confusion matrix along with the actual classifications in order to determine the accuracy of the model.

```
# crossvalidate the model using the remaining 30% of dataset
predict_CrossVal <- predict(model, crossval_set)
confusionMatrix(crossval_set$classe, predict_CrossVal)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    A    B    C    D    E
##          A 1673    1    0    0    0
##          B   11 1124    4    0    0
##          C    0    8 1018    0    0
##          D    0    0   13  951    0
##          E    0    0    0    4 1078
##
## Overall Statistics
##
##                Accuracy : 0.993
##                  95% CI : (0.9906, 0.995)
##     No Information Rate : 0.2862
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.9912
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: A Class: B Class: C Class: D Class: E
## Sensitivity            0.9935   0.9921   0.9836   0.9958   1.0000
## Specificity            0.9998   0.9968   0.9984   0.9974   0.9992
## Pos Pred Value         0.9994   0.9868   0.9922   0.9865   0.9963
## Neg Pred Value         0.9974   0.9981   0.9965   0.9992   1.0000
## Prevalence             0.2862   0.1925   0.1759   0.1623   0.1832
## Detection Rate         0.2843   0.1910   0.1730   0.1616   0.1832
## Detection Prevalence   0.2845   0.1935   0.1743   0.1638   0.1839
## Balanced Accuracy      0.9966   0.9944   0.9910   0.9966   0.9996
```

This model yielded a 99.3% prediction accuracy which proved that the model is robust and adequete to predict new datasets.

USING THE MODEL FOR PREDICTIONS

The testing datasets, pml-testing.csv was then loaded into R and similar preprocessing to be carried out as before with the pml-training.csv. The model will be used to predict the classifications of the 20 results of this new data.

```r
# apply the same pre-procesiing to the final testing datasets, pml-testing.csv
pml_test <- read.csv("./pmldata/pml-testing.csv", na.strings= c("NA",""," "))
pml_test_NAs <- apply(pml_test, 2, function(x) {sum(is.na(x))})
pml_test_clean <- pml_test[,which(pml_test_NAs == 0)]
pml_test_clean <- pml_test_clean[8:length(pml_test_clean)]

# predict the classe of the test datasets
predict_Test <- predict(model, pml_test_clean)
predict_Test
```

```
##  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20
##  B  A  B  A  A  E  D  B  A  A  B  C  B  A  E  E  A  B  B  B
## Levels: A B C D E
```

CONCLUSION

With the abundance of inexpensive datasets collected from mutliple health devices such as Jawbone Up, Nike FuelBand, Fitbit, etc, it is possible to accurately predict how well a person is preforming an excercise using a relatively simple machine learning model.

APPENDIX

R Libraries

The following libraries were used throughout:

```
library(caret)
library(corrplot)
library(kernlab)
library(knitr)
library(randomForest)
```

Data sources

Both he training (pml-training.csv) and testing (pml-testing.csv) files were downloaded directly from the internet and stored in a created file folder (./pmldata) in the working directory.

```r
# check if a data folder exists; if not then create one
if (!file.exists("data")) {dir.create("data")}

# data URL and destination file folder
Url1 <- "https://d396qusza40orc.cloudfront.net/predmachlearn/pml-training.csv"
destfile1 <- "./pmldata/pml-training.csv"
Url2 <- "https://d396qusza40orc.cloudfront.net/predmachlearn/pml-testing.csv"
destfile2 <- "./pmldata/pml-testing.csv"

# download the file and note the time
download.file(Url1, destfile = destfile1)
download.file(Url2, destfile = destfile2)
```